

札幌市ITイノベーション研究会御中

超高速開発が企業システムに革命を起こす



2015／03／06

本間 峰一

一般社団法人ICT経営パートナーズ協会

<http://www.ictm-pa.jp/>

(株)ほんま コンサルティング事業部

<http://www.homma-consulting.jp/>

E-Mail: m.homma@mbf.nifty.com

自己紹介

■ 本間 峰一（ほんま みねかず）

1958年生まれ。

株式会社ほんま コンサルティング事業部 経営コンサルタント

一般社団法人ICT経営パートナーズ協会 理事

電気通信大学電気通信学部卒業。日本電気（株）コンピュータシステムの営業および営業企画職（中部支社、製造システム事業部）を経て、1994年富士総合研究所（みずほ総合研究所の前身）入社。

みずほ総合研究所(株)コンサルティング部 上席主任コンサルタント（2012年6月退職し、経営コンサルタントとして独立）

収益性向上、マーケティング強化、生産管理、SCM構築、情報化方針立案などを支援目的とした経営コンサルティングを担当（約110社）。

中小企業診断士、情報処理技術者（システムアナリスト、システム監査技術者、プロジェクトマネージャ、アプリケーションエンジニア）、日本生産管理学会会員、日本経営情報学会会員。

主な著書：

『受注生産に徹すれば利益はついてくる』（単著、日刊工業新聞社）

『コストダウンが会社をダメにする』（単著、日刊工業新聞社）

『超高速開発が企業システムに革命を起こす』（共著、日経BP社）

『SAP革命』（編著、日本能率協会マネジメントセンター）

『失敗しないERP導入ハンドブック』（編著、日本能率協会マネジメントセンター）

『サプライチェーンマネジメントがわかる本』（編著、日本能率協会マネジメントセンター）

『IT失敗学の研究』（共著、日経BP社）

『製造業の戦略的情報化マニュアル』（監修、通産資料調査会）ほか

はじめに：情報システム開発費用の高騰が止まらない

最近出会ったケース

- ◆ N社：年商300億円⇒見積額 3億円（MWサポ-ト停止）
- ◆ T社：年商200億円⇒見積額 5億円（ERP導入）
- ◆ J社：年商120億円⇒見積額 6億円（スクラッチ開発）
- ◆ M社：年商 70億円⇒実績額 3億円（スクラッチ開発）

M社は本番開始3年でユーザ苦情が多発し、作り直しを決断

安く導入できるはずだったのにERPパッケージは高くついた

最大の問題はERPパッケージは日本ではそのままでは使えないこと

- ◆ 取引先のわがままを受け入れることで生き残ってきた企業も多く、ERPパッケージにあわせることで要求に対応できなくなると取引自体が成立しなくなる可能性がある

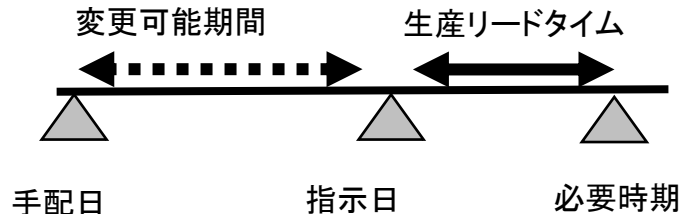
⇒日本のほとんどの企業は受注ビジネスをしているのでこのことは死活問題

- ◆ ERPパッケージが、取引単位、取引形態、売上計上時期、相場変動、ロットトレース、返品・変更・取消処理などの複雑な業務処理手順をサポートしていない
- ◆ 納品荷姿個数、重量、長さなど複数単位での製品や材料の管理が必要な企業もあり、複数単位に対応できないERPパッケージでは適切な業務運営ができない
- ◆ ERPパッケージでは細かな管理分類データ項目が設定しにくいので、独自の経営数値分析や業務数値分析の実施には限界がある（ex. 何の工事や保守作業？）
- ◆ オフコンや汎用コンピュータ上に開発した独自システムを使っていた企業では、機能が限定されるERPパッケージ利用だと使い勝手に支障が生じる可能性がある
- ◆ 独自のビジネスモデルにより商売をしている企業では、そもそもそのモデルにあったERPパッケージが存在しない

ノンカスタマイズのパッケージ利用なのにカスタマイズ設計が発生し、システム開発費用はかえって高くついた

MRP生産管理パッケージ（ERPの原点）の問題点

- MRP計算は各部品の必要時期を個別に計算し、生産リードタイムをさかのぼった時点で生産指示する



- このことにより次のような効果が得られる

- ① 前倒し生産を防げるので余分な在庫の発生を抑制できる
- ② 発注日までは変更が発生しても生産はしないので(変更可能期間)、変更による混乱を抑制できる

- ところが、

- ① 手配時にインプットする必要時期の精度が悪く、必要時期の変更が頻繁に起こる
- ② 生産リードタイムの精度が低く、必要時期に調達できないことがある
- ③ 製造工程の能力と負荷状況によって生産リードタイムは変化するがそれが考慮されていない

- この状態ではMRPは機能しないので、MRPの仕組みを殺し、手配日に一律リードタイムで生産指示する

- 結果的に実際の納期は生産管理担当が個別に手作業で調整管理するしかなくなる

- ◆ MRPは計画通りに調達するためのシステムで、計画がぶれては意味をなさない
- ◆にもかかわらず、大半の生産管理パッケージはMRPをベースにしている

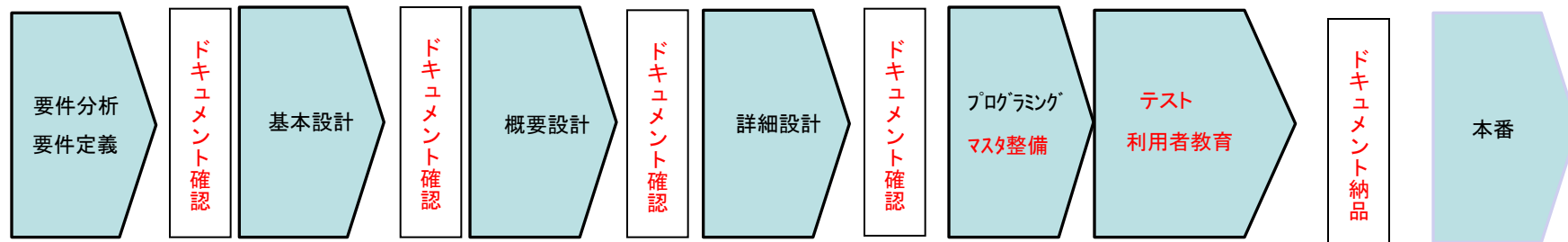
昔のパッケージは個別仕様に対応できた

1980年代のパッケージ	現在のパッケージ(ERPほか)
• どこかの会社用に開発したものをテンプレート化してパッケージと称していた • パッケージのソースコードは原則ユーザに提供された • ユーザ企業もしくはソフト会社が自由にカスタマイズが可能だった • パッケージベンダがサポート停止しても利用可能 • パッケージソフトコードは買取りが基本で、ソフト保守という概念はほとんどなかった • バージョンアップはほとんどなかった(当時のOSも上位互換を保証していた) 個別カスタマイズが前提	• ベンダ独自の設計思想に基づき開発されている • パッケージのソースコードは非公開で、ブラックボックス化されている • カスタマイズはベンダしかできない(アドオンも制限有) • パッケージベンダがなくなったり、サポート停止すると原則継続利用は難しくなる • ソフト保守料が必要(ソフト保守の内容はあいまいなことが多い) • ベンダ都合やOS対応のバージョンアップがあり、莫大な対応費用がかかることもある ノンカスタマイズが前提

**現在のパッケージは昔と違って開発自由度が大きく制限されたため
下手にカスタマイズしようとするとかえって多額の費用がかかる**

かといってウォーターフォール型スクラッチ開発は高くつく

ウォーターフォール型開発



- 各工程が完了するごとにドキュメント(成果物)による確認作業を実施してから次の工程に移る
- システム開発業者の赤字転落リスクを軽減できるので、大手のSI業者を中心に本手法が定着した



ウォーターフォール型開発が高くなりやすい理由

- ① 下流工程での仕様変更が相次いで、想定よりも開発規模が膨れ上がりやすい
- ② 図で赤字にした部分のユーザ作業が遅れやすく、その分の追加費用が発生する
- ③ SES主体の開発はプロジェクトを解散すると再結成が難しいため、工期遅れ時にSES体制維持費用が必要になる(ウォーターフォール開発は工期遅れが付きもの)
- ① ドキュメント作成、変更、確認作業に必要以上に工数と費用をかけて請求するSI業者がいる(半分以上がドキュメント関係費用といったケースもある)
- ② 従来はベンダが負担していた経費増加リスクをユーザに転嫁しやすくなった(確認したドキュメント内容と違うので追加費用が必要など)

設計段階で仕様確定できるというのは幻想にすぎない

実際に開発に入ってみると

- 設計書やフロー図をみるだけでIT素人の経営者や利用者がシステムイメージを把握することはほとんど不可能に近い
- 利用者は自分が属人的に処理してきたことや手作業、メモ作業は情報システム構築とは無関係と思っている人が多く、設計ヒアリングではでてきにくい
- 利用者は実際に画面操作してはじめて抜けている機能や足りない処理に気付くのが普通
- 利用者は実際に触り始めないとマスタ情報やデータ定義の重要性を理解できない
- 集計分類は実際に集計画面をみてから急に追加要求が増加する傾向がある
- 処理ロジックは利用者も設計段階では勘違いしていることがある

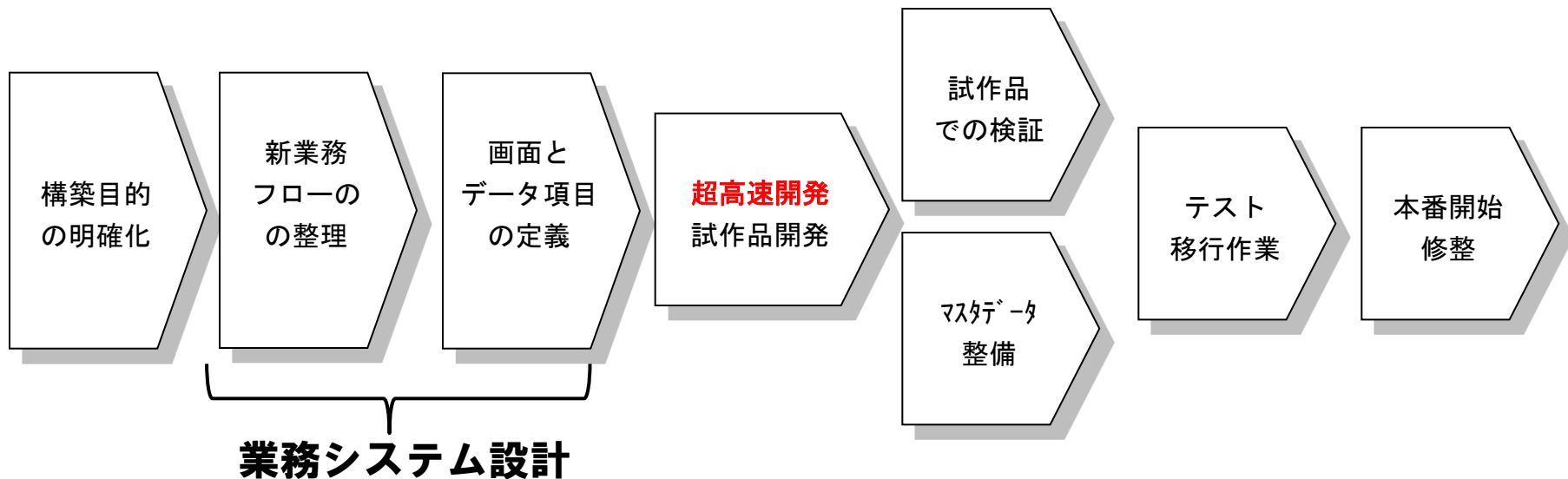


- ◆ これでは予算(当初契約額)以内でシステムが完成すること自体が疑わしい
- ◆ もしも予算(契約額)内で終わったとしたら、当初見積額が過大だっただけ？



設計工程に時間をかけるよりも曖昧な内容でいいから実際に動くシステムを早く作って利用者に触ってもらって検討するのが本来の開発アプローチ

進化型プロトタイプ（試作品）開発を提案します



進化型プロトタイプシステム開発の進め方

- ◆ 超高速開発ツールを使って実際に動作する試作品をさっさと作ってしまう
- ◆ ここで作った試作品をそのまま修整していくことで、本番システムに仕立て上げていく
- ◆ **業務システム設計では基本となる業務フローを整理するだけで、細かな仕様検討は試作品を操作・検証しながら実施する。どうせ変更されるとわかっているので、ムダなドキュメントは作らない**
- ◆ 実際に動く試作品をベースに仕様検証するので、利用者による動作確認が容易にできる

- **製品開発、建築の世界では試作品を用いて仕様検討するのが当たり前**
- **優秀なコンサルタントがリードしないと検証作業が発散して収拾がつかなくなる**

進化型プロトタイプ開発はなぜ普及してこなかったのか

- ① 進化型プロトタイプを支援する本格的な開発ツールがなかった
 - 試作品開発のプログラム工数がかかっているとは本番開発とかわらない
 - 試作品として画面イメージだけをみせても利用者は理解しづらい

⇒システム開発現場には製品開発における3Dプリンタのようなツールがなかった
- ② SIベンダが要求発散リスクを嫌がった
 - 実物に接することで利用者から追加要求が相次ぐ可能性がある
 - 設計書（ドキュメント）による仕様確認と共存しにくい
 - 超高速開発ツールを使うとベンダが開発費用を稼げない
- ③ 業務フローを整理するSE（システムエンジニア）が不足している
 - パッケージの台頭で、業務フロー整理よりも機能設定にSEの仕事がシフト
 - 下請受託・派遣主体の開発業者ばかりで、プログラマはいるがSEがいない
 - 外注依存分業体制が進み、全体業務をみることのできる人が減った

超高速開発ツールが実用化されたことで、若手エンジニアや、往年のCOBOLエンジニアでも、進化型プロトタイプに耐えられる試作品を素早く作って仕様確認することができるようになりました

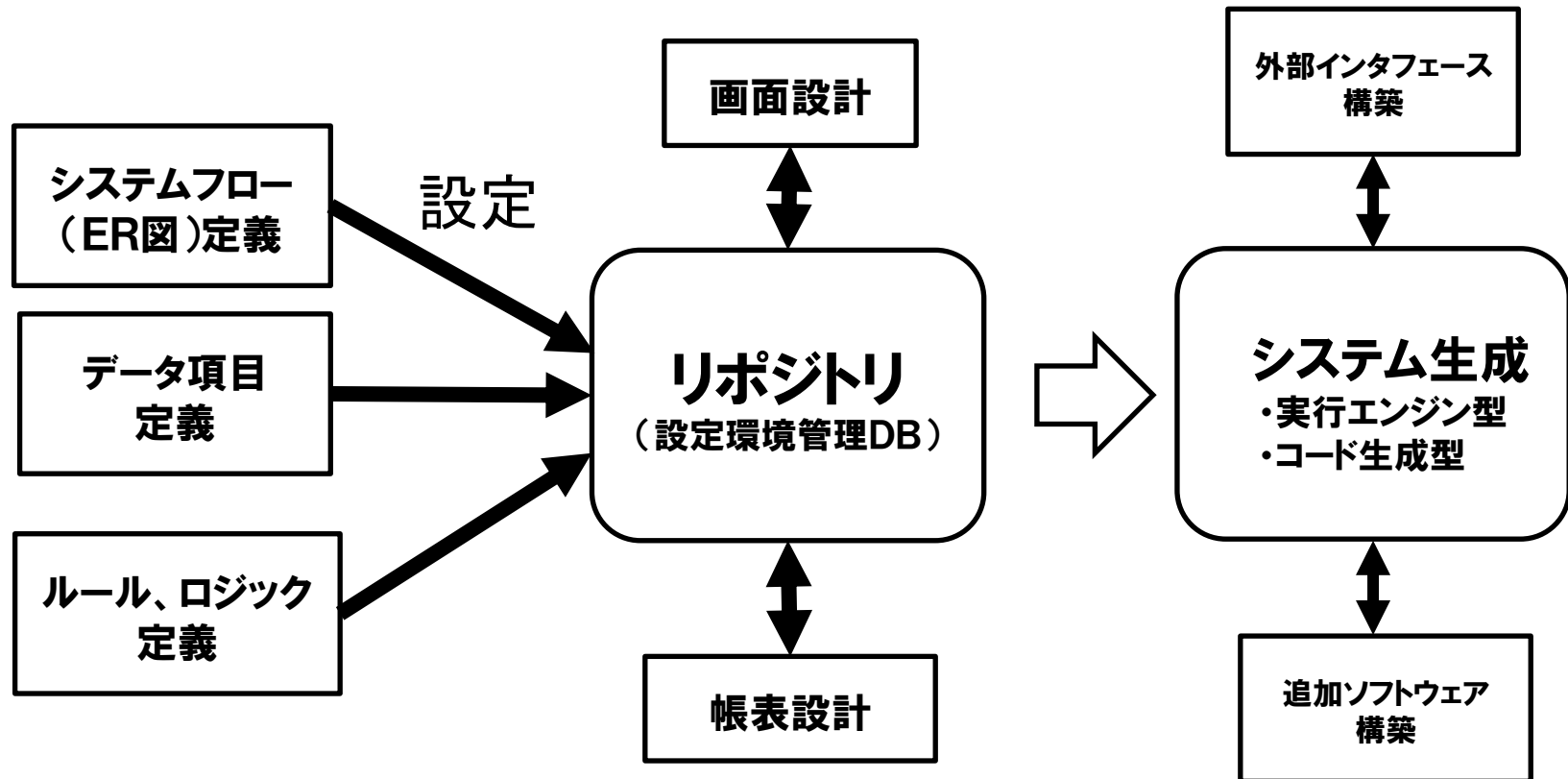
『超高速開発ツール』とは？

超高速開発ツールとは、業務処理用の情報システム開発作業を効率化して、短期間かつ少ない開発工数でシステム開発ができるようにするための開発ツールの総称。情報システム開発費用が高騰していることや、パッケージを利用したシステム開発では機能面や使い勝手面での不満が大きいことから、主に自社開発企業を中心に導入が進んでいる

- プログラミングが必要ないので、専門技術者でなくても開発できるうえに、開発工数を削減することができます
- 単体テストが必要ないので、テスト効率が大幅にアップします
- リポジトリ(設定環境管理DB)で仕様を一元管理するので、システム変更やシステム保守にも柔軟に対応できます
- どのツールも基本的なアプローチは似ているので、設計内容は流用することができます
- 画面レイアウトツールを内蔵しているので、試作品画面を簡単に作ることができます

- ◆ エンジニアのいない中小企業でも自社開発とシステムメンテが少人数で容易にできるようになるため、SIベンダに頼ったシステム導入をしなくてもすむ
- ◆ パッケージ利用によってわかりにくくなった情報システムによる差別化や顧客サービス強化を真剣に考える経営 風土を復活させることができる
- ◆ 情報システム開発におけるITコーディネータやコンサルタントの役割が重要になる

超高速開発ツールによるシステム生成の考え方



ウォーターフォール(WF)との生産性の比較

		WF	アジャイル	xRAD	アジャイル /WF	xRAD/WF
総費用/JFS	平均	112.19	135.45	40.7	1.21	0.36
	係数	28.2	57.65	6.4		
工数/JFS	平均	1.28	2.15	0.48	1.68	0.37
	係数	0.44	1.6	0.26		
工期/JFS	平均	0.31	0.24	0.1	0.77	0.32
	係数	0.13	0.04	0.03		

・費用・工数・工期ともに超高速開発はウォーターフォール法の3分の1である

備考

JFS:JUAS Function Scale。画面数と帳票数から規模を換算した値。既存WFのデータを基に画面数+帳票数×2/3 を算出(ユーザー発注者が明確に判るのは画面数、帳票数である)。また、係数はグラフにプロットしたときの傾きを示す。xRADは係数が小さいので、規模による生産性の変動が少ない。

備考: 表はJUAS様作成

主な業務アプリ開発用超高速開発ツール

No	ベンダ名	開発ツール / メソドロジー	開発国	得意適用領域	システム生成
1	株式会社ジャズミンソフト	Wagby	日本	業務アプリ	コード生成
2	ジェネクス・ジャパン株式会社	GeneXus	ウルグアイ	業務アプリ	コード生成
3	ケン・システムコンサルティング株式会社	MDFrame/X	日本	業務アプリ	コード生成
4	株式会社BlueMeme	Outsystems Platform	ポルトガル	業務アプリ	コード生成
5	キャノンソフトウェア株式会社	Web Performer	日本	業務アプリ	コード生成
6	TIS株式会社	Xenlon～神龍	日本	業務アプリ	コード生成
7	株式会社エヌ・ティ・ティ・データ	TERASOLUNA	日本	業務アプリ	コード生成
8	マジックソフトウェア・ジャパン株式会社	Magic xpa Application Platform	イスラエル	業務アプリ	実行エンジン
9	株式会社アプストウェブ	コンテキサー	日本	業務アプリ	実行エンジン
10	株式会社アトリス	PEXA	日本	業務アプリ	実行エンジン
11	サピエンス・ジャパン株式会社	Sapiens	イスラエル	業務アプリ	実行エンジン
12	株式会社ケーエムケーワールド	A's Style	日本	業務アプリ	実行エンジン
13	株式会社ファルコン	TALON	日本	業務アプリ	実行エンジン
14	住友電工情報システム株式会社	楽々FRAMEWORK II	日本	業務アプリ	部品
15	株式会社GCT研究所	iRYSHA	日本	業務アプリ	部品

出典: 超高速開発が企業システムに革命を起こす

代表的な超高速開発ツール「Sapiens」

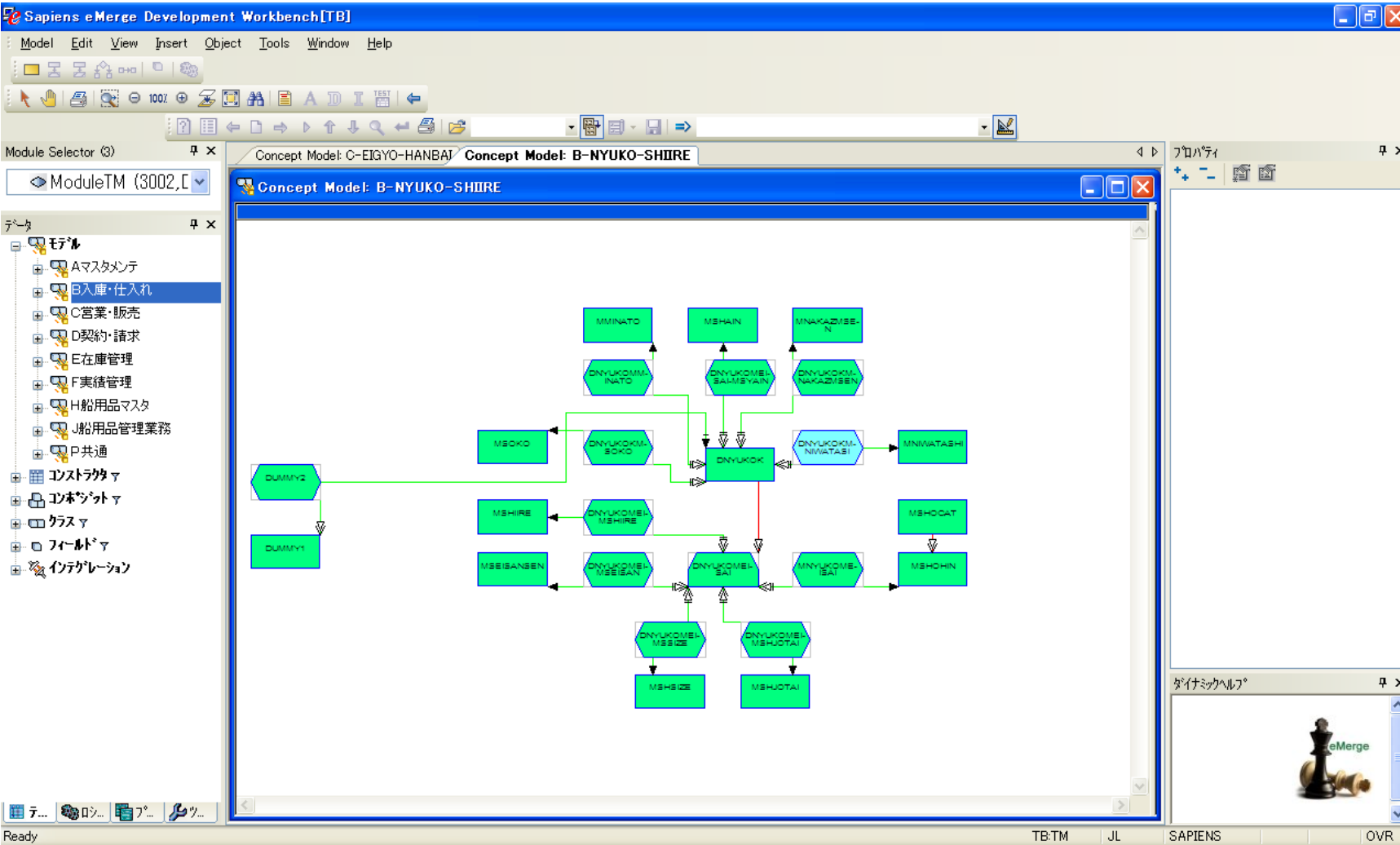
- ◆ イスラエルで生まれた販売システムを効率的に構築するための開発プラットフォーム
 - 約20年前にイスラエルのワイツマン研究所で誕生
 - 当初はIBM製のメインフレーム上のみで動いていたが、Window版やLinux版もリリースされるようになった。
 - イスラエルの政府機関の情報システムの多くはサピエンスを利用している
 - 日本や欧州では大企業での自社開発ツールとしての導入が進んでいる
- ◆ 一般的には超高速開発ツールといわれているが、単なるツールではなく個別開発（スクラッチ開発）用のシステム基盤である
 - 販売システムの動作に必要な仕組みがあらかじめ組み込まれている
 - リポジトリに動作設定して利用する（データ定義、画面、ルール、ロジックなど）
 - DBは外部の汎用DBを利用する
 - ポジティブシンキング機能により例外処理の記述が必要ない
 - 進化型プロトタイプを作って仕上げていく形でのシステム開発に適している
 - イメージ的にはAccessの開発方法に似ている
 - 政府機関、大企業中心に導入されていただけに細かい機能設定が可能

業務システム設計さえできあがれば、短時間でシステム実装できます

サピエンスのオートデモ

http://www.tojolab.com/sapidemo/html/TMSL_DEMO_main.html

Sapiensの開発ステップ ①ER図の作成



Sapiensの開発ステップ ②データ定義

Sapiens eMerge Development Workbench[TB]

ファイル(F) 表示(V) アクション(A) ツール(T) ウィンドウ(W) ヘルプ(H)

Module Selector (3) ModuleTM (3002,C)

Concept Model: C-EIGYO-HANBAI Concept Model: B-NYUKO-SHIRE コンセプト-2120

データ モデル

- Aマスタメンテ
- B入庫・仕入れ
- C営業・販売
- D契約・請求
- E在庫管理
- F実績管理
- H船用品マスタ
- J船用品管理業務
- P共通
- コンストラクタ
- コンセプト
- クラス
- フィールド
- インテグレーション

Concept Model: B-NYUKO-SHIRE

コンセプト-2120

コンセプト: DNYUKOK

タイプ: Entity

ステータス: Implementation

テンプレート: 10689

アクション

範囲セットID: SYS

複雑性: Very Complicated

フィールド	データ型	サイズ	小数桁数	ドメイン
NYKFUNENO	船番号	4900 Fixed	6	NYKFUNE
Character				
NYKNYUKOYMD	入港日	4901 Date	10	NYKNYUK
NYKIREYMD	入庫日	4902 Date	10	NYKIREYM
NYKINPYMD	入力日	4903 Date	10	NYKINPYM
NAKAZMSENC	仲積船コード	3400 Fixed	4	
NAKAZMSENNM	仲積船	3401 Fixed	24	
MINATOC	港コード	3500 Fixed	4	
MINATONM	港	3501 Fixed	20	
SOKOCD	倉庫コード	3700 Fixed	3	
SOKONM	倉庫	3701 Fixed	30	
SOKOCHIKI	倉庫地域	3702 Fixed	1	SOKOCHIKI
NYKLOTNO	入庫番号	4970 Fixed	12	NYKLOTNO
NYKTMIOFLG	積卸料フラグ	4904 Fixed	1	NYKTMIOFLG
NYKNJURYOTTL	ネット重量合計	4927 PackedDecimal	14	2 NYKNJURYOTTL

ドメイン(D)

プロパティ

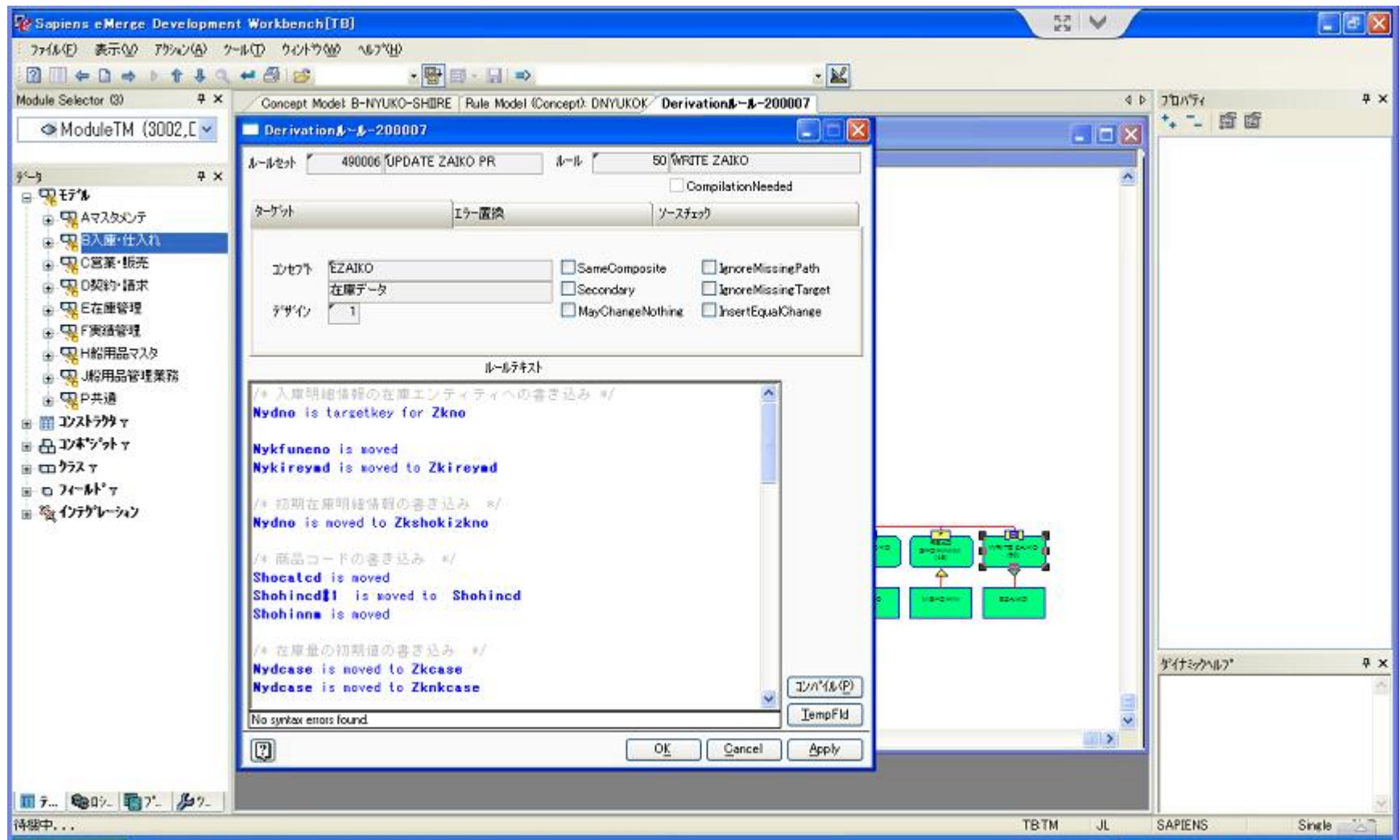
コンセプトフィールドプロパティ

- 名前: 英語 SOKONM, 日本語 倉庫
- 全般: フィールド番号 3701, サイズ 30, ドメイン Entity Reference
- データ型プロパティ: データ型 Fixed
- 文字: AlphaNumeric ☐, AlphaBetic ☐, SecondLang ☒, Mixed ☒, SCL-Mixed ☐, PureDBCS ☐
- 数値: 小数点置 MayBeNegat ☐, NotZero ☐, Quantity ☐
- 日付: ChangeDate ☐, CreateDate ☐
- アタラク: DB2 SQL-Time ☐, Virtual ☐
- 役割: NotInClass ☐, NotInOperation ☐

ダイナミックヘルプ

Ready TB-TM JL SAPIENS Single OVR

Sapiensの開発ステップ ③ロジック定義



Sapiensで作成した画面の例

製品登録-320010

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

内外作判断システム

LOGIN ID: TEST 権限: 一般社員
社員番号: 9001 社員名: TEST

製品新規登録

バージョン: 1.0

会社コード: C002 会社名略称: TEST2 入力日付: 2014/07/22

機種: ICTMP-001 仕様:
図番: ICTMP-001 版数:
名称: 研修装置用部品

数量/月(ロット): 50 製品重量(kg): 100.00 キロ単価: 151

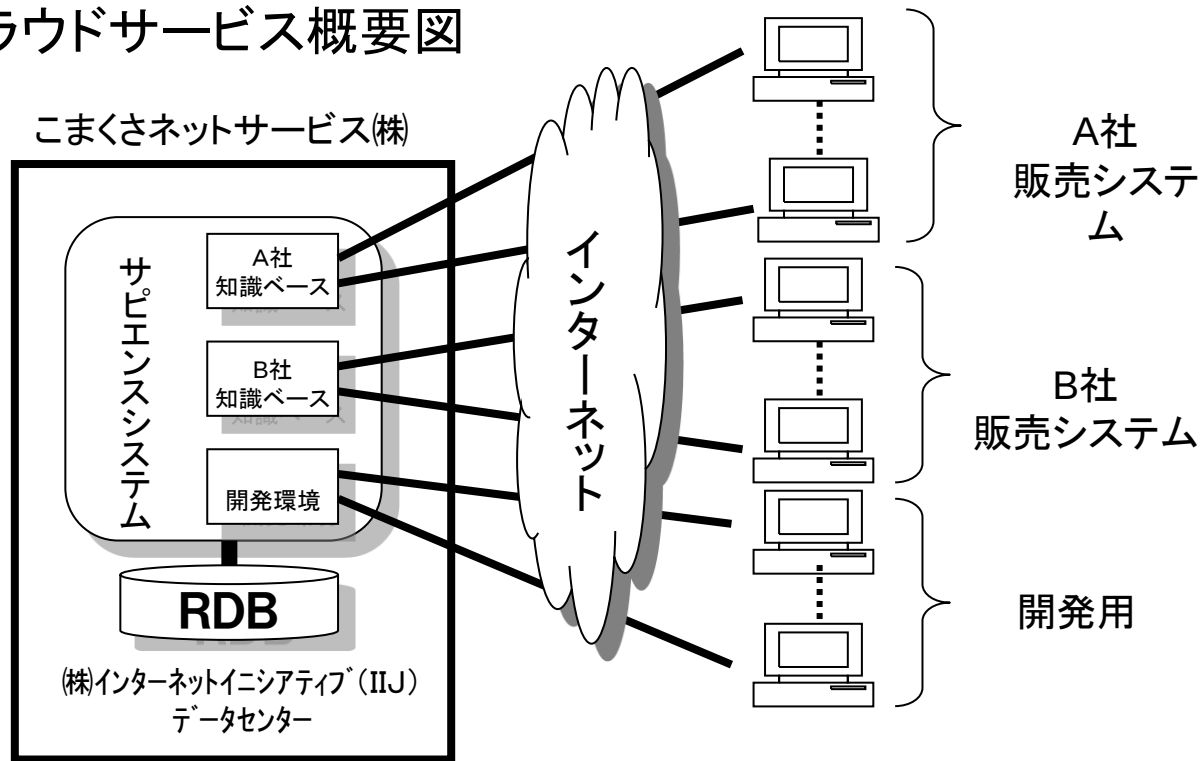
材料費(合計): 9172 材料費(経費): 276 材料費(掛率: %): 3.00 スクラップ(比率: 合計): 0
加工費(合計): 4200 加工費(経費): 630 加工費(掛率: %): 15.00 運賃(合計): 0
保管費(経費): 429 保管費(掛率: %): 3.00 金型割掛(合計): 0 見積金額: 15,136
横持ち費(経費): 429 横持ち費(掛率: %): 3.00 その他(合計): 0 スループット(粗利): 4,764

順番	機種	仕様	図番	版数	名称	構成数	有支/無支	有償支給品番	区分	製品重量
1	板金部品	鉄	ICTMP-001-001	1	ベース	1			M: 自社調達材	2.00C
2	曲げ	作業	ICTMP-001-002		曲げ作業	0				0.00C

こまくさネットサービスによるSapiensクラウドサービス

こまくさネットサービス(株)は「Sapiens」の開発・実行クラウド環境を中小企業向けに安くレンタル提供する会社です（開発：月3万円～ 実行：月15万円～）

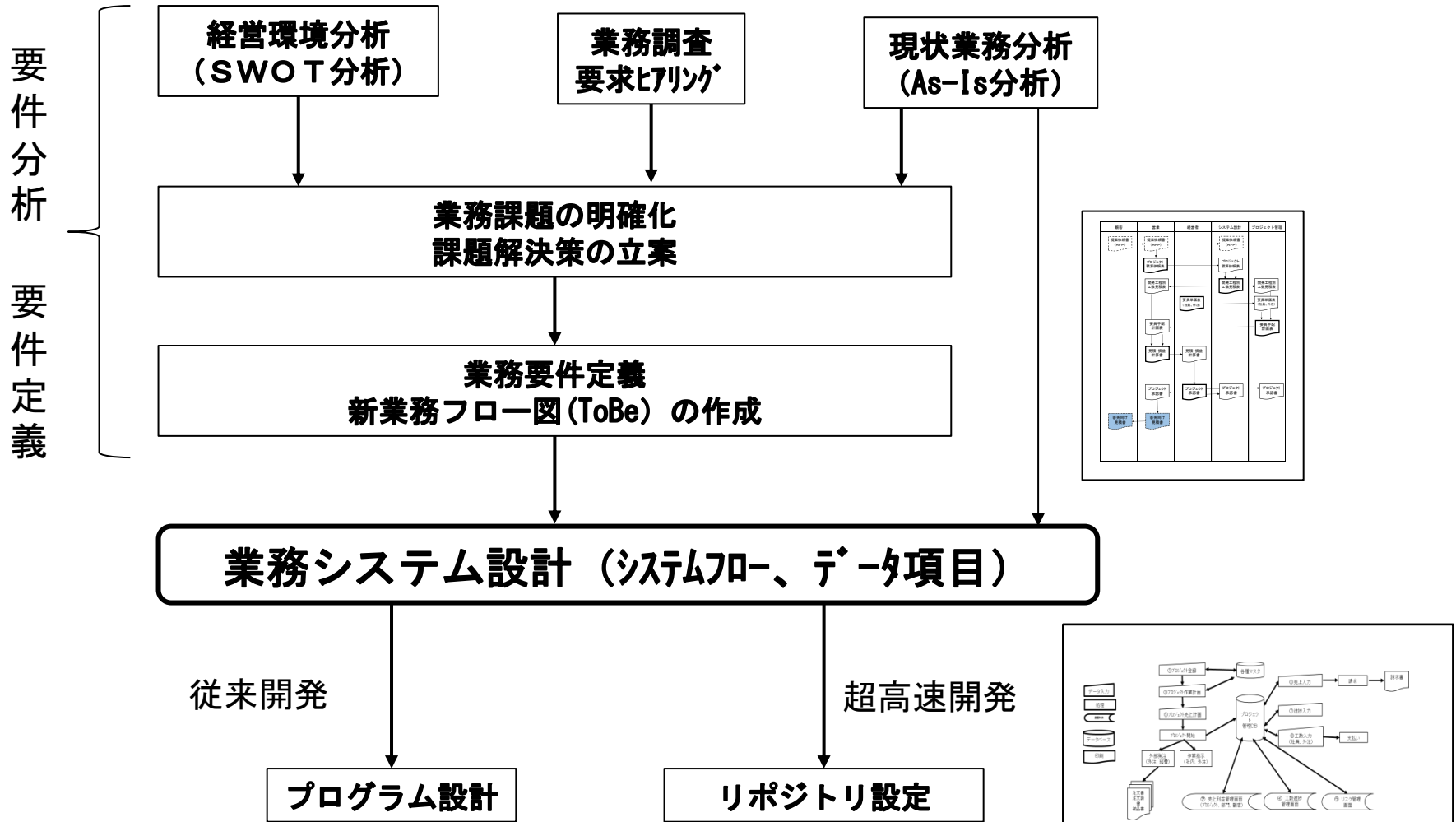
クラウドサービス概要図



こまくさネットサービスのサービスを使えば、試しにSapiensを使ってみたり、Sapiensを使ってシステム検討用の試作品を作成することができます（契約は月単位）

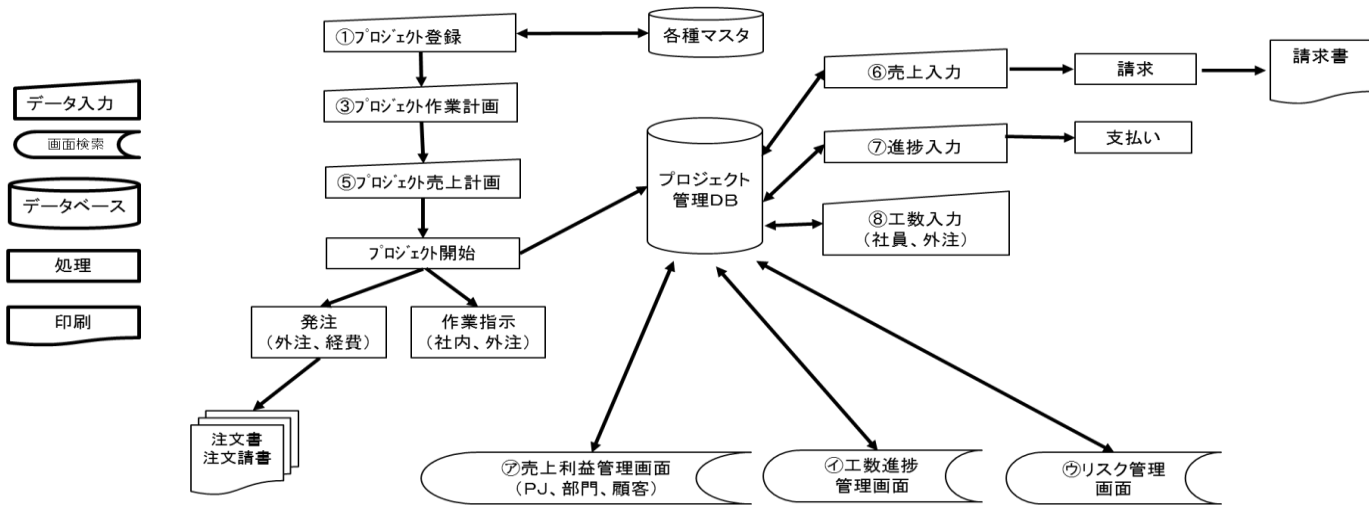
⇒ 使ってみたい方は小生までご連絡ください

業務システム設計の位置づけ



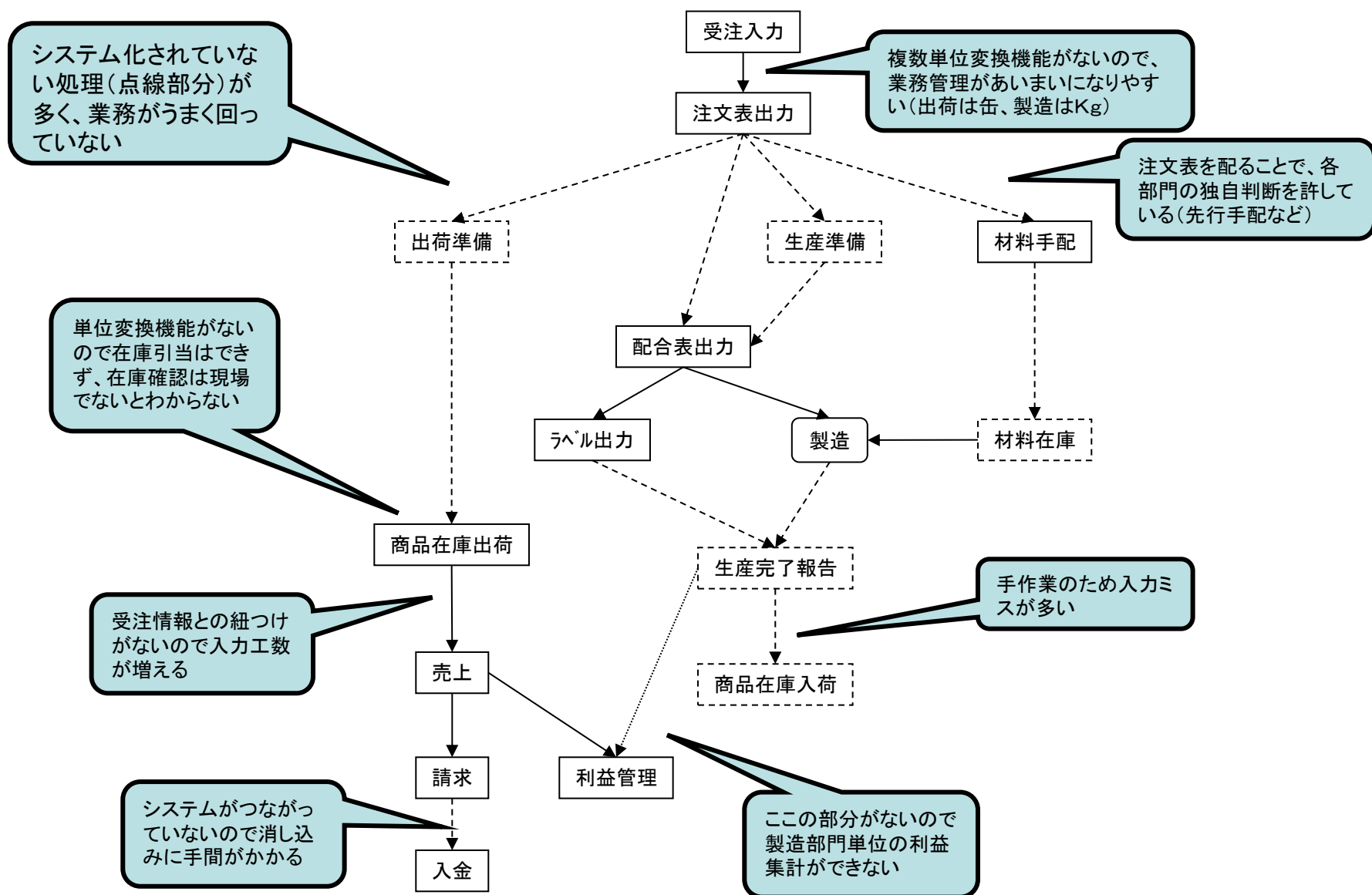
システムフロー図とデータ項目の例

サンプルフローチャート



画面分類	番号	フィールド名 (日本語)	KEY	マスタ 利用	プル ダウン	チェック ボックス	計算値	引用先画面	備考	種別	桁数	少数
基本入力	1	プロジェクト番号			○					数値	5	
	2	プロジェクト名称						①		文字	20	
	3	プロジェクト分類						①		文字	10	
	4	顧客名						①		文字	60	
	5	PJ予定工数合計						④		数値	6	1
	6	PJ見積額						④		数値	10	0
	7	PJ総予定原価						④		数値	10	0
	8	PJ基準利益						④		数値	10	0
	9	PJ基準利益率						④		%	3	1
	10	契約金額							契約値	数値	10	0
	11	契約金額利益率							(契約金額 - PJ総予定原価 - 今季共通配賦経費) ÷ 契約金額	%	3	1
	12	超過時間単価							契約値	数値	6	1
	13	控除時間単価							契約値	数値	6	1
	14	契約工数							PJ予定工数合計と異なる場合に入力	数値	6	1
	15	契約差工数					○		契約工数 - PJ予定工数合計	数値	6	1
	16	売上予定金額合計					○		明細合計 (契約金額との一致確認用)	数値	10	1
	17	生産進捗率合計					○		明細合計 (100%入力の確認用)	%	3	1
明細入力	1	入力月								日付	2	
	2	売上予定金額								数値	10	0
	3	生産進捗予定率							%入力	%	3	1
	4	生産予定金額					計算値		契約金額 × 生産予定進捗率	数値	10	0

業務システム設計ができていなかったシステム事例（S社）



試作品検証開発方式におけるコンサルタントの役割

従来の流れ



- コンサルタントが行った要件定義は無視され、SI会社がパッケージ前提ではじめからやり直す
- パッケージ前提の場合のコンサルタント主体の要件定義には限界がある(パッケージの詳細機能を知らない)
- パッケージ前提なのにコンサルタントが口出し過ぎるとカスタマイズが増えやすい

これからの流れ



コンサルタント（中小企業診断士、ITコーディネータ）が加わって業務システム設計した上で試作品開発するアプローチが定着すれば、ユーザ企業がSI会社に丸投げする日本独特の悪習もなくせます

★ RFP(Request For Proposal:提案依頼書) RFQ(Request For Quotation):見積依頼書

ご清聴ありがとうございました